

日 本 国 特 許 庁

JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2000年 5月18日

CERTIFIED COPY OF
PRIORITY DOCUMENT

出 願 番 号
Application Number:

特願2000-152665

出 願 人
Applicant(s):

株式会社日立製作所

App. No. 09/855,681

(703) 684-1120

MATTINGLY, STANGER & MALUR, P.C.

1800 DIAGONAL ROAD

SUITE 370

ALEXANDRIA, VIRGINIA 22314

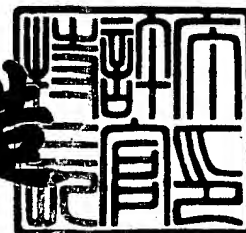
DKt. No. ASA-1000

BEST AVAILABLE COPY

2001年 4月20日

特許庁長官
Commissioner,
Japan Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 K00000391

【提出日】 平成12年 5月18日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 11/28

【請求項の数】 7

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 鶴ヶ崎 俊博

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マルチプロセスの表示方法及び装置並びに表示プログラムを格納した記録媒体

【特許請求の範囲】

【請求項 1】

プログラムを実行するマルチプロセッサの実行状態を表示するための表示方法であって、前記プログラムを構成するプロセスを所定の図形で表示し、前記プロセス間の関係を示すよう前記図形間に所定の表示を行い、前記プロセスの動作状態に連動して前記図形を特徴表示することを特徴とするマルチプロセスの表示方法。

【請求項 2】

請求項 1 記載のマルチプロセスの表示方法であって、前記プロセス間の関係は、プロセス間の親子関係又は兄弟関係であることを特徴とするマルチプロセスの表示方法。

【請求項 3】

請求項 1 記載のマルチプロセスの表示方法であって、前記プロセスの動作状態は、プロセスの生成・開始・再開・停止・終了の少なくとも 1 つであることを特徴とするマルチプロセスの表示方法。

【請求項 4】

請求項 1 記載のマルチプロセスの表示方法であって、各プロセスの詳細情報を前記図形と同一画面に表示することを特徴とするマルチプロセスの表示方法。

【請求項 5】

請求項 4 記載のマルチプロセスの表示方法であって、前記図形の特徴表示と前記各プロセスの詳細情報とは、連動して強調表示されることを特徴とするマルチプロセスの表示方法。

【請求項 6】

プログラムを実行するマルチプロセッサの実行状態を表示するための表示装置であって、前記プログラムを構成するプロセスを所定の図形で表示し、前記プロセス間の関係を示すよう前記図形間に所定の表示を行う第一の表示手段と、前記

プロセスの動作状態に連動して前記図形を特徴表示する第二の表示手段とを備えたことを特徴とするマルチプロセスの表示装置。

【請求項 7】

プログラムを構成するプロセスを所定の図形で表示し、前記プロセス間の関係を示すよう前記図形間に所定の表示を行い、前記プロセスの動作状態に連動して前記図形を特徴表示することによりプログラムを実行するマルチプロセッサの実行状態を表示することを特徴とするマルチプロセスの表示プログラムを格納した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マルチプロセスの表示方法及び装置並びに表示プログラムを格納した記録媒体に係り、特にプログラムを構成するプロセスの動作状態の変化を、プログラムの動作と連動しながら視覚的な観点でデバッグするためのマルチプロセスプログラムのデバッグ方法及び装置に関するものである。

【0002】

【従来の技術】

従来からデバッグを支援するために様々な方法が考えられている。例えば、特開平 5 - 2 3 3 3 2 3 号においては、並列プログラムのデバッグを対話的に効率良く行うためのデバッグ方法が開示されている。また、特開平 8 - 1 8 5 3 4 0 号においては、並列プログラムを可視化するためのツールである並列プログラム可視化方法が開示されている。しかし、これらの従来技術では、並列プログラムを構成するプロセスの親子、及び兄弟関係を視覚的に特徴表示する機能は有していない。また、プロセスの生成、開始、再開、停止、及び終了というプロセスの動作状態を特徴表示する機能は有していない。尚、特開平 1 1 - 1 6 1 5 1 4 号においては、複数のプロセスで構成させるプログラムのデバッグ方法を示しているが、プログラムの動作と連動しながら、プログラムを構成するプロセスのデバッグを行える機能は有していない。

【0003】

【発明が解決しようとする課題】

上記従来技術では、並列プログラムを構成するプロセスの動作状態の変化に応じたデバッグ方法が配慮されておらず、例えば、並列プログラムを構成するプロセスが増大した場合、プロセスの生成、開始、再開、停止、及び終了のようなプロセスの動作に着目してプログラムのデバッグを行うことが困難という問題があった。また、並列プログラムの動作と連動したデバッグ方法が配慮されておらず、例えば、動作プロセスと、終了プロセスとの親子、及び兄弟関係も重要なデバッグ作業を行うための情報（以下、デバッグ情報と呼ぶ）となるが、そのデバッグ情報に着目したデバッグが困難という問題があった。

【0004】

本発明の目的は、マルチプロセスプログラムのデバッグを効率的に行うことを可能とするマルチプロセスの表示方法及び装置並びに表示プログラムを格納した記録媒体を提供することにある。

【0005】

【課題を解決するための手段】

上記目的は、マルチプロセスプログラムを構成する複数のプロセスの動作状態に関連付けたデバッグ情報を、プログラムの動作と連動しながら特徴表示する点、また動作プロセスと終了プロセスとの関係を視覚的に区別表示する点に着目することにより達成される。このような点に着目することにより、複数のプロセスで構成されるマルチプロセスプログラムのデバッグを効果的に可能とするデバッグ支援方法及び装置を提供できる。

【0006】

より具体的には、プログラム（マルチプロセスプログラム、並列プログラムを含む）を実行するマルチプロセッサの実行状態を表示するための表示方法であって、前記プログラムを構成する複数のプロセスをそれぞれ所定の図形（例えば、矩形）で表示（以下、プロセスボックスとも呼ぶ）し、前記プロセス間の関係を示すよう前記図形間に所定の表示（例えば、線による連結表示等の視覚的に表現する表示）を行い、前記プロセスの動作状態に連動して前記図形を特徴表示（例

例えば、プロセスボックスの枠線を、動作プロセスは実線、終了プロセスは破線) することで、プロセスの動作状態を視覚的に表現したもの(以下、プロセス関連グラフとも呼ぶ)により達成する。ここで、プロセス間の関係とは、プロセスの親子又は兄弟関係であっても良い。尚、プロセスの動作状態とは、例えばプロセスの生成・開始・再開・停止・終了等を示す。また、プロセス関連グラフは、プログラムの動作に連動しながら、プロセスボックスの特徴表示状態を変化させるものであっても良い。さらに、プロセスの詳細な動作状態(プログラム名称やソースコード位置の情報)を簡略化して知ることが出来るようにするために、プロセス関連グラフと、プロセスの詳細な動作情報(以下、プロセス詳細情報と呼ぶ)を同一画面に表示し、プロセス関連グラフとプロセス詳細情報は、どちらか片方の情報から、あるプロセスの強調表示操作を行った場合でも、プロセス関連グラフと、プロセス詳細情報の両情報のプロセス情報を連動して強調表示するようにしたものであっても良い。

【0007】

尚、上記目的を達成するためには、上述した機能を実現するプログラム若しくはプログラムを格納した記録媒体であっても良い。

【0008】

【発明の実施の形態】

以下、本発明の一実施形態を図面に基づいて詳細に説明する。

【0009】

図1は、マルチプロセスプログラムのデバッグ方法の構成図である。デバッグ作業を行うプログラム(以下、デバッグプログラムと呼ぶ)100、デバッガシステム部110、及び表示装置150で構成される。デバッグプログラム100は、プログラムを構成する複数のプロセス群(以下、プロセスをデバッグプロセスと呼ぶ)101で構成される。デバッガシステム部110は、デバッグプログラム100の実行を制御、及び実行を監視するデバッガエンジン部120と、デバッグプログラム100の動作状態を、表示装置150に表示するデバッガ情報表示部140で構成させる。デバッガエンジン部120は、デバッグプログラム100を、デバッグプロセス101単位に実行を制御、及び実行を監視するデバ

ッグ実行処理部 1 2 1 と、デバグエンジン部 1 2 0 より、デバグ情報表示部 1 4 0 へ、デバグプロセス 1 0 1 の動作状態を伝達するために使用するプロセス動作情報 1 3 1 を、プロセス動作情報ファイル 1 3 0 に出力するプロセス動作情報出力部 1 2 2 で構成される。デバグプロセス 1 0 1 の動作状態に変化が生じた場合、デバグ実行処理部 1 2 1 が検知し、プロセス動作情報出力部 1 2 2 に制御を移す。プロセス動作情報出力部 1 2 2 は、プロセス動作情報ファイル 1 3 0 に、プロセス動作情報 1 3 1 を出力する。プロセス動作情報 1 3 1 の詳細は図 2 で説明する。プロセス動作情報監視部 1 4 1 は、デバグプロセス 1 0 1 の動作に連動しながら、表示装置 1 5 0 の表示状態を変化させるために、プロセス動作情報ファイル 1 3 0 へのプロセス動作情報 1 3 1 出力の有無を調べる。プロセス動作情報 1 3 1 が出力されている場合、プロセス動作情報監視部 1 4 1 は、プロセス動作情報取得部 1 4 2 へ制御を移し、プロセス動作情報取得部 1 4 2 では、プロセス動作情報ファイル 1 3 0 から、プロセス動作情報 1 3 1 を入力し、プロセス情報テーブル組立部 1 4 3 へ制御を移す。プロセス情報テーブル組立部 1 4 3 は、入力したプロセス動作情報 1 3 1 から、デバグプロセス 1 0 1 を管理するプロセス情報テーブルを作成及び更新し、プロセス情報表示部 1 4 4 に制御を移す。プロセス情報テーブルの詳細は図 3 で説明する。プロセス情報表示部 1 4 4 は、プロセス関連グラフ表示部 1 4 5 と、プロセス詳細情報表示部 1 4 6 で構成され、プロセス情報テーブル組立部 1 4 3 で作成、及び更新されたプロセス情報テーブルを参照し、プロセス関連グラフ 1 5 1 とプロセス詳細情報 1 5 2 を、表示装置 1 5 0 に表示する。

【 0 0 1 0 】

図 2 は、プロセス動作情報 1 3 1 のデータ構成図である。プロセス動作情報 1 3 1 は、プロセス状態情報 2 0 1 と、プロセス詳細情報 2 0 2 で構成される。プロセス状態情報 2 0 1 は、デバグプロセス 1 0 1 の動作状態情報を管理し、プロセス詳細情報 2 0 2 は、デバグエンジン部 1 2 0 が、デバグプロセス 1 0 1 を、内部的に管理するために付加する番号（以下、ノード番号と呼ぶ）や、プロセス番号、プログラム名称、及び停止位置のソースファイル情報など、プロセス状態情報 2 0 1 に応じた情報を管理する。プロセス動作情報 2 1 0 は、デバッ

グプロセス 1 0 1 が新たに生成された場合のプロセス動作情報 1 3 1 の出力例である。ノード番号 (0) 2 1 2、プロセス番号 (1 0) 2 1 3 のデバッグプロセスから、ノード番号 (1) 2 1 4、プロセス番号 (2 0) 2 1 5 のデバッグプロセスが生成されたことを意味する。プロセス動作情報 2 2 0 は、ファイルの動作が開始された場合のプロセス動作情報 1 3 1 の出力例である。ノード番号 (1) 2 2 2、プロセス番号 (2 0) 2 2 3 のデバッグプロセスで、プログラム名称 (P r o g 2 _ 0) 2 2 4 の実行が開始されたことを意味する。プロセス動作情報 2 3 0 は、ブレークポイントなどの要因により、動作を停止した場合のプロセス動作情報 1 3 1 の出力例である。ノード番号 (2) 2 3 2、プロセス番号 (2 1) 2 3 3 のデバッグプロセスは、ソースファイル (f u n c t i o n . c) 2 3 4 の行番号 (1 4 6 行目) 2 3 5 で停止したことを意味する。プロセス動作情報 2 4 0 は、デバッグプロセスの動作を再開した場合のプロセス動作情報 1 3 1 の出力例であり、ノード番号 (0) 2 4 2、プロセス番号 (1 0) 2 4 3 のデバッグプロセスは、動作を再開したことを意味する。プロセス動作情報 2 5 0 は、デバッグプロセスの動作が終了した場合のプロセス動作情報 1 3 1 の出力例であり、ノード番号 (3) 2 5 2、プロセス番号 (3 0) 2 5 3 のデバッグプロセスは、動作が終了したことを意味する。

【 0 0 1 1 】

図 3 は、図 1 に示すプロセス情報テーブル組立部 1 4 3 により作成、及び更新され、プロセス関連グラフ表示部 1 4 5、及びプロセス詳細情報表示部 1 4 6 で参照するプロセス情報テーブル 3 0 0 の詳細である。デバッグプロセス情報 3 1 0、プロセスボックス座標位置情報 3 4 0、子供デバッグプロセス情報 3 2 0、兄弟デバッグプロセス情報 3 3 0 で構成される。デバッグプロセス情報 3 1 0 は、ノード番号情報 3 1 1、プロセス番号情報 3 1 2、プログラム名称情報 3 1 3、及び動作状態情報 3 1 4 を管理する。プロセスボックス座標位置情報 3 4 0 は、プロセス情報テーブル 3 0 0 で管理しているデバッグプロセスを、プロセスボックスとしてプロセス関連グラフに、表示した場合の座標情報として、プロセスボックスの x 軸情報 3 4 1、y 軸情報 3 4 2、幅情報 3 4 3、及び高さ情報 3 4 4 を管理する。子供デバッグプロセス情報 3 2 0 は、プロセス情報テーブル 3 0

0で管理しているデバッグプロセスより生成されたデバッグプロセス（以下、子プロセスと呼ぶ）のプロセス情報テーブルポインタ321を管理し、兄弟デバッグプロセス情報330は、プロセス情報テーブル300で管理しているデバッグプロセスと、兄弟の関係にあるデバッグプロセス（以下、兄弟プロセスと呼ぶ）のプロセス情報テーブルポインタ331を管理する。

【0012】

図4は、プロセス動作情報ファイル130からプロセス動作情報131を入力し、表示装置150に表示するデバッグ情報表示部140の処理手順をフローチャートにしたものである。まず、プロセス動作情報監視部141は、プロセス動作情報ファイル130に、プロセス動作情報131が出力されているかを調べる（ステップ411）。プロセス動作情報131が出力されている場合、プロセス動作情報取得部142に制御を移す。プロセス動作情報取得部142は、プロセス動作情報ファイル130から、プロセス動作情報131を入力し（ステップ421）、プロセス情報テーブル組立部143へ制御を移す。プロセス情報テーブル組立部143では、プロセス動作情報131に管理されているプロセス状態情報より、デバッグプロセスの動作状態を調べる（ステップ431）。プロセス状態情報が、デバッグプロセス生成時は、プロセス情報テーブル作成処理（ステップ432）、デバッグプロセスの動作開始、再開、停止、及び終了時は、プロセス情報テーブル更新処理（ステップ433）を行い、プロセス情報表示部144に制御を移す。プロセス情報テーブル作成処理432の詳細な実施例は図5に、プロセス情報テーブル更新処理433の詳細な実施例は図6に記載する。プロセス情報表示部144は、プロセス情報テーブルをキーにして、プロセス関連グラフ表示部145で、プロセス関連グラフ表示用のデータ作成処理442を、プロセス詳細情報表示部146で、プロセス詳細情報表示用のデータ作成処理443をデバッグプロセスの数分繰り返し処理し（ステップ441～444）、表示装置150に制御を移す。プロセス関連グラフ表示データ作成処理442、及びプロセス詳細情報表示データ作成処理443の詳細な実施例は図7に記載する。

【0013】

図5は、プロセス情報テーブル作成処理432の詳細な実施例である。図2で

説明したプロセス動作情報 2 1 0 を入力例として説明する。まず、新しいデバッグプロセス 1 0 1 を生成したデバッグプロセス（入力例の場合、ノード番号 0 ～ 2 1 2、プロセス番号 1 0 ～ 2 1 3 のデバッグプロセス。以下、親プロセスと呼ぶ）を管理しているプロセス情報テーブルの検索 5 0 1 を行う。検索方法は、親プロセスのノード番号（入力例の場合、ノード番号 0 ～ 2 1 2）と、プロセス情報テーブルに管理しているノード番号情報との一致判定 5 0 2 で行う。親プロセスを管理しているプロセス情報テーブルを検索できたら、新しく生成されたデバッグプロセス（入力例の場合、ノード番号 1 ～ 2 1 4、プロセス番号 2 0 ～ 2 1 4 のデバッグプロセス。以下、生成プロセスと呼ぶ）のプロセス情報テーブルを作成 5 0 3 し、親プロセスのプロセス情報テーブルで管理している子供デバッグプロセス情報に登録する。同時に、兄弟デバッグプロセス情報も登録し、プロセス情報表示部～1 4 4 に制御を移す。

【 0 0 1 4 】

図 6 は、プロセス情報テーブル更新処理 4 3 3 の詳細な実施例である。図 2 のブレイクポイントなどの要因により、動作を停止した場合のプロセス動作情報 2 3 0 を入力例として説明する。まず、動作の停止したデバッグプロセス（入力例の場合、ノード番号 2 ～ 2 3 2、プロセス番号 2 1 ～ 2 3 3 のデバッグプロセス）を管理しているプロセス情報テーブルの検索 6 0 1 を行う。検索方法は、デバッグプロセスのノード番号（入力例の場合、ノード番号 2 ～ 2 3 2）と、プロセス情報テーブルに管理しているノード番号情報との一致判定 5 0 2 で行う。デバッグプロセスを管理しているプロセス情報テーブルを検索出来たら、デバッグプロセスの動作状態を、プロセス動作情報 2 3 0 のプロセス状態情報 2 3 1 で判定し（ステップ 6 0 3）、入力例のプロセス動作情報 2 3 0 のプロセス状態情報 2 3 1 は動作停止を表しているため、デバッグプロセスを管理しているプロセス情報テーブルの動作状態情報に、動作停止情報を設定し（ステップ 6 0 6）、プロセス情報表示部 1 4 4 に制御を移す。同様に、プロセス動作情報 2 2 0 を入力例とした場合、動作状態情報に動作開始情報を設定（ステップ 6 0 4）、プロセス動作情報 2 4 0 を入力例とした場合、動作状態情報に動作再開情報を設定（ステップ 6 0 5）、プロセス動作情報 2 5 0 を入力例とした場合、動作状態情報に動

作終了情報を設定し（ステップ607）、プロセス情報表示部144に制御を移す。

【0015】

図7は、プロセス情報表示部144の詳細な実施例である。まず、プロセス情報テーブル組立部143で作成、及び更新された、プロセス情報テーブルを入力し（ステップ701）、全てのデバッグプロセスのデバッグ情報表示処理が完了したかを判断する（ステップ702）。全てのデバッグプロセスのデバッグ情報表示処理が完了した場合、表示装置150へ制御を移す。プロセス関連グラフ表示部145では、デバッグプロセスをプロセスボックスとして表示するために、プロセスボックスの表示位置情報（x軸、y軸、幅、及び高さ）の計算を行い（ステップ703）、プロセス情報テーブルのプロセスボックス座標位置情報に、それぞれの計算値を設定する（ステップ704）。その後、デバッグプロセスの動作状態（生成、開始、再開、停止、及び終了）と連動してプロセスボックスを特徴表示するため、プロセス情報テーブルの動作状態情報を参照し（ステップ705）、動作状態に応じたプロセスボックス表示用画面データの作成（ステップ710、ステップ711、ステップ712、ステップ713、又はステップ714）を行い、プロセス詳細情報表示部146へ制御を移す。プロセス詳細情報表示部146は、プロセス詳細情報表示用の画面データの作成を行う（ステップ443）。

【0016】

図8は、表示装置150の同一画面に表示されたプロセス関連グラフ151と、プロセス詳細情報152の表示例である。まず、プロセス関連グラフ151に表示されるプロセスボックス（801、802、803、及び804）には、デバッグプロセスを示す情報として、プロセス番号（ノード番号）のデバッグプロセス情報を表示する。例として、プロセスボックス801のデバッグプロセス情報810は、プロセス番号10（ノード番号0）のデバッグプロセスである事を示す。更に、プロセスボックス801は、プロセスボックス802と線により連結表示811していることで、プロセスボックス802で示されるデバッグプロセスを生成した親プロセスとなる。また、プロセスボックス804とも、線によ

り連結表示 8 1 2 していることで、プロセスボックス 8 0 4 で示されるデバッグプロセスを生成した親プロセスとなる。これで、プロセスボックス 8 0 2 と、プロセスボックス 8 0 4 間は、兄弟プロセスであることが分かる。プロセスボックス 8 0 2 は、プロセスボックス 8 0 3 と線により連結表示 8 1 3 されていることで、プロセスボックス 8 0 3 で示されるデバッグプロセスを生成した親プロセスである。プロセスボックス 8 0 3、及びプロセスボックス 8 0 4 は、線により連結表示されているプロセスボックスが無い場合、子プロセスの生成は行っていない。更に、プロセスボックス 8 0 1 のデバッグプロセスは、プロセスボックスの枠線を太線により特徴表示 8 4 1 していることで、動作再開状態のデバッグプロセスであることを示し、プロセスボックス 8 0 2 と、プロセスボックス 8 0 4 のプロセスは、プロセスボックスの枠線を細線により特徴表示 (8 4 2、8 4 3) していることで動作停止状態、プロセスボックス 8 0 3 のプロセスは、プロセスボックスの枠線を点線により特徴表示 8 4 3 していることで実行終了状態と、デバッグプロセスの動作状態に連動した表示を行う。プロセス詳細情報 1 5 2 は、行の情報として、デバッグプロセスの詳細な動作情報 (8 2 1、8 2 2、8 2 3、及び 8 2 4) を表示し、列の情報として、ノード番号 8 3 1、プログラム名称 8 3 2、プロセス番号 8 3 3、動作状態 8 3 4、及び停止位置 8 3 5 のデバッグ情報を表示する。

【 0 0 1 7 】

図 9 は、プロセス関連グラフ 9 1 1 からの、特定デバッグプロセスのデバッグ情報選択状態 (以下、強調表示と呼ぶ) 操作 9 1 0 と、プロセス詳細情報 9 2 1 からの、特定デバッグプロセスのデバッグ情報強調表示操作 9 2 0 の手順である。プロセス関連グラフ 9 1 1 からの強調表示操作 9 1 0 では、マウスなどにより、プロセス関連グラフ 9 1 1 から、プロセス番号 2 0 (ノード番号 1) のデバッグプロセスのデバッグ情報強調表示操作 9 1 2 が行われた場合、マウスのポイント位置と、プロセス情報テーブルのプロセスボックス座標位置情報で管理している座標情報から、マウスのポイント位置に、プロセスボックスとして表示されているデバッグプロセスを管理しているプロセス情報テーブルを検索する (ステップ 9 1 3)。検索したプロセス情報テーブルより、強調表示操作の行われたデバ

ッグプロセスのノード番号は1であることを取得でき（ステップ914）、取得したノード番号をキーにして、プロセス詳細情報916に表示しているプロセス詳細情報から、ノード番号の一致するプロセス詳細情報の行の情報を検索する（ステップ915）。プロセス情報テーブル検索処理（ステップ913）で検索したプロセス情報テーブルのプロセスボックス座標位置情報から、プロセスボックスの強調表示処理を行い（ステップ930）、プロセス詳細情報検索（ステップ915）で検索した行の情報を基に、プロセス詳細情報の強調表示処理を行う（ステップ931）。これで、プロセス関連グラフ941のノード番号1のデバッグプロセスを示すプロセスボックス943、及びプロセス詳細情報942のノード番号1のデバッグプロセスを示すプロセス詳細情報944が連動して強調表示され、表示装置940に表示する。次に、プロセス詳細情報921からの強調表示操作920では、マウスなどにより、プロセス詳細情報921から、プロセス番号20、ノード番号1のデバッグプロセスのデバッグ情報強調表示操作922が行われた場合、マウスのポイント位置から、行の情報としてデバッグプロセスの詳細情報を表示している文字列の取得を行い（ステップ923）、取得した文字列からノード番号を取得する（ステップ924）。取得したノード番号をキーにして、強調表示操作の行われたデバッグプロセスを管理している、プロセス情報テーブルを検索する（ステップ925）。プロセス関連グラフからの強調表示操作910と同様、プロセス関連グラフ941のノード番号1のデバッグプロセスを示すプロセスボックス943、及びプロセス詳細情報942のノード番号1のデバッグプロセスを示すプロセス詳細情報944が連動して強調表示され、表示装置940に表示される。

【0018】

このように本実施例によれば、マルチプロセスプログラムを構成する複数のプロセスの動作状態に関連付けたデバッグ情報を、プログラムの動作と連動しながら特徴表示することにより、プロセスの生成、開始、再開、停止、及び終了のような、プロセスの動作に着目したプログラムのデバッグを行うことができる。また、プロセス関連グラフと、プロセス詳細情報を同一画面に表示し、プロセス関連グラフとプロセス詳細情報は、連動して強調表示することにより、プロセスの

詳細な動作状態（プログラム名称やソースコード位置の情報）を簡略化して知ることが出来る。

【 0 0 1 9 】

【発明の効果】

本発明によれば、各プロセッサにおけるマルチプロセスプログラムの実行状態の詳細情報を可視化することができ、デバッグの効率化を支援することができる。

【図面の簡単な説明】

【図 1】

マルチプロセスプログラムのデバッグ方法の構成図である。

【図 2】

プロセス動作情報のデータ構成図である。

【図 3】

プロセス情報テーブル詳細である。

【図 4】

デバッグ情報表示部の処理手順をフローチャートにしたものである。

【図 5】

プロセス情報テーブル作成処理の詳細な実施例である。

【図 6】

プロセス情報テーブル更新処理の詳細な実施例である。

【図 7】

プロセス情報表示部の詳細な実施例である。

【図 8】

表示装置に表示されたプロセス関連グラフと、プロセス詳細情報の表示例である。

【図 9】

プロセス関連グラフとプロセス詳細情報が連動して強調表示される手順である。

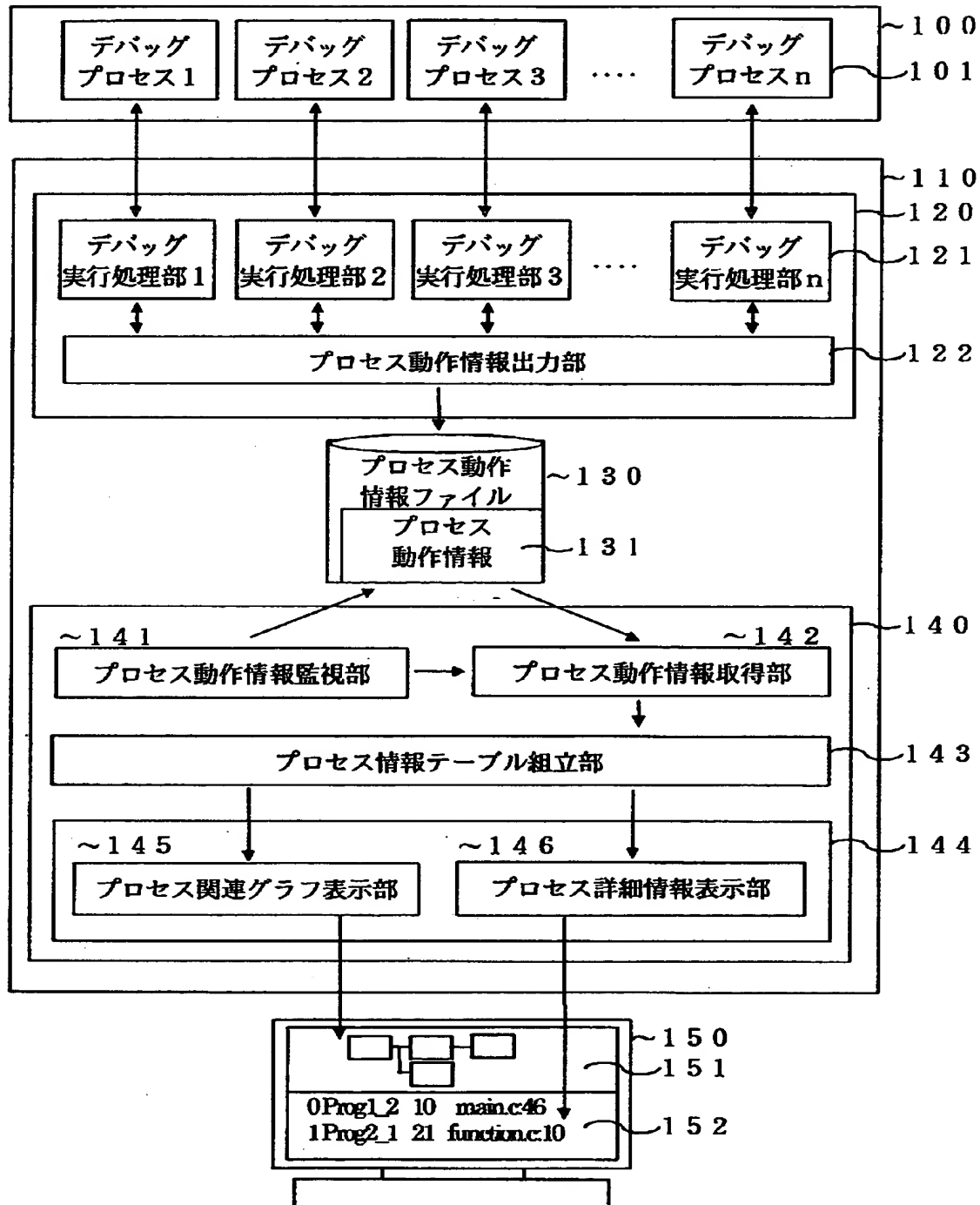
【符号の説明】

- 1 0 0 デバッグプログラム
- 1 1 0 デバッガシステム部
- 1 2 0 デバッガエンジン部
- 1 3 0 プロセス動作情報ファイル
- 1 4 0 デバッガ情報表示部
- 1 5 0 表示装置
- 2 0 1 ~ 2 5 3 プロセス動作情報出力例
- 3 0 0 プロセス情報テーブル
- 3 1 0 デバッグプロセス情報
- 3 2 0 プロセスボックス座標情報
- 3 3 0 子供デバッグプロセス情報
- 3 4 0 兄弟デバッグプロセス情報
- 4 1 1 ~ 4 4 4 デバッガ情報表示部の処理手順
- 5 0 1 ~ 5 0 4 プロセス情報テーブル作成処理
- 6 0 1 ~ 6 0 7 プロセス情報テーブル更新処理
- 7 0 1 ~ 7 1 4 プロセス情報表示部
- 8 0 1 ~ 8 0 4 プロセスボックス
- 8 1 0 デバッグプロセス情報
- 8 1 1 ~ 8 1 3 プロセス親子及び兄弟関係の特徴表示
- 8 2 1 ~ 8 2 4 デバッグプロセスの詳細な動作情報（行情報）
- 8 3 1 ~ 8 3 5 デバッグプロセスの詳細な動作情報（列情報）
- 8 4 1 ~ 8 4 4 プロセスボックスの特徴表示
- 9 1 0 プロセス関連グラフからの強調表示操作
- 9 2 0 プロセス詳細情報からの強調表示操作
- 9 4 0 プロセス関連グラフとプロセス詳細情報の強調表示

【書類名】 図面

【図 1】

図 1



【図2】

図2

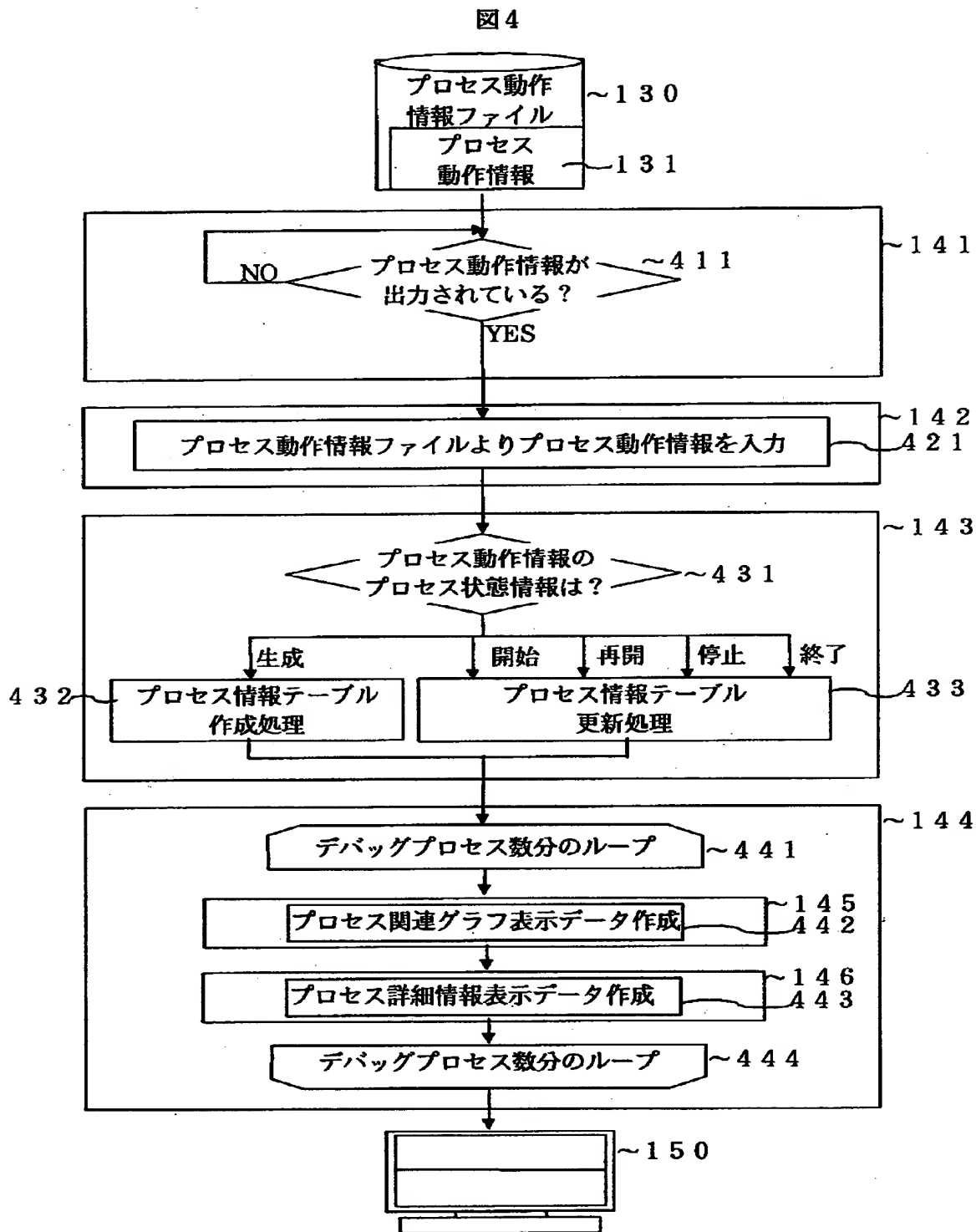
~201		~202		
131~	プロセス状態情報	プロセス詳細情報		
~211	~212	~213	~214	~215
プロセス状態情報	親ノート番号	親プロセス番号	ノート番号	プロセス番号
210~ 生成	0	10	1	20
~221	~222	~223	~224	
プロセス状態情報	ノート番号	プロセス番号	プログラム名称	
220~ 開始	1	20	Prog2_0	
~231	~232	~233	~234	~235
プロセス状態情報	ノート番号	プロセス番号	ソースファイル	行番号
230~ 停止	2	21	function.c	146
~241	~242	~243		
プロセス状態情報	ノート番号	プロセス番号		
240~ 再開	0	10		
~251	~252	~253		
プロセス状態情報	ノート番号	プロセス番号		
250~ 終了	3	30		

【図 3】

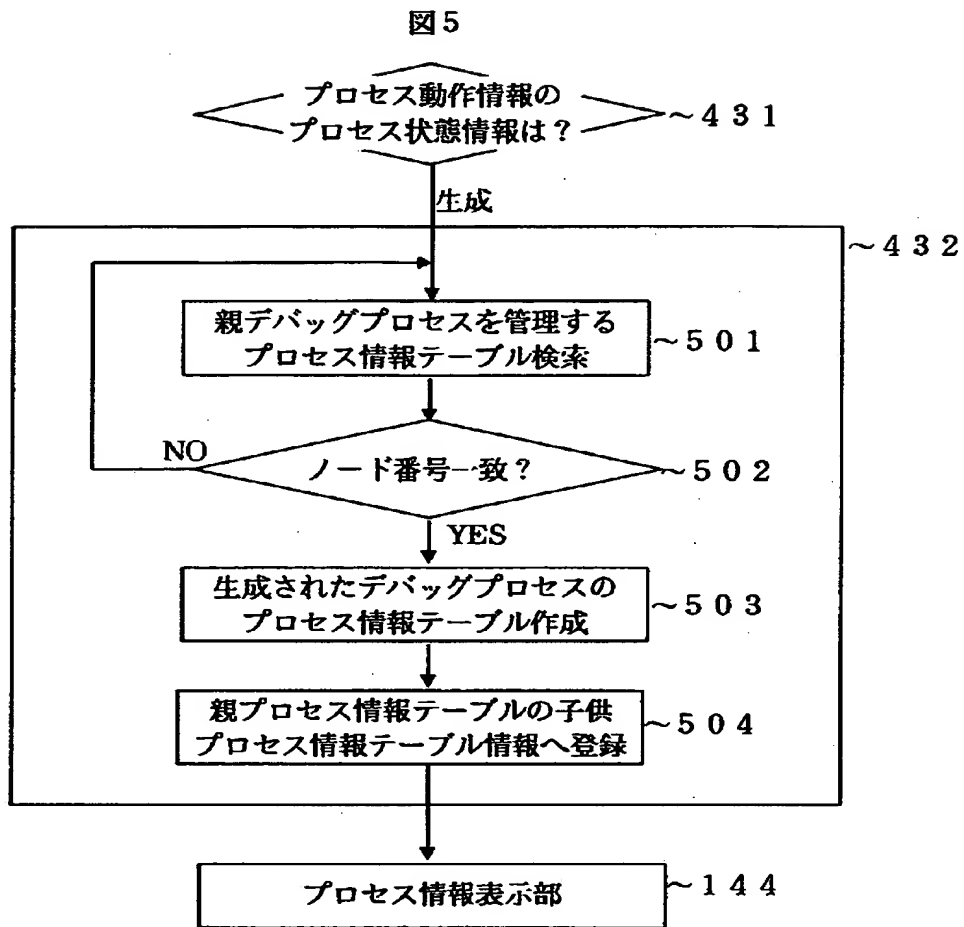
図 3

～ 3 0 0			
3 1 0～	デバッグプロセス情報	ノード番号情報	～ 3 1 1
		プロセス番号情報	～ 3 1 2
		プログラム名称情報	～ 3 1 3
		動作状態情報	～ 3 1 4
3 4 0～	プロセスボックス 座標位置情報	x 軸情報	～ 3 4 1
		y 軸情報	～ 3 4 2
		幅情報	～ 3 4 3
		高さ情報	～ 3 4 4
3 2 0～	子供デバッグプロセス 情報	子供プロセス情報ポインタ	～ 3 2 1
3 3 0～	兄弟デバッグプロセス 情報	兄弟プロセス情報ポインタ	～ 3 3 1

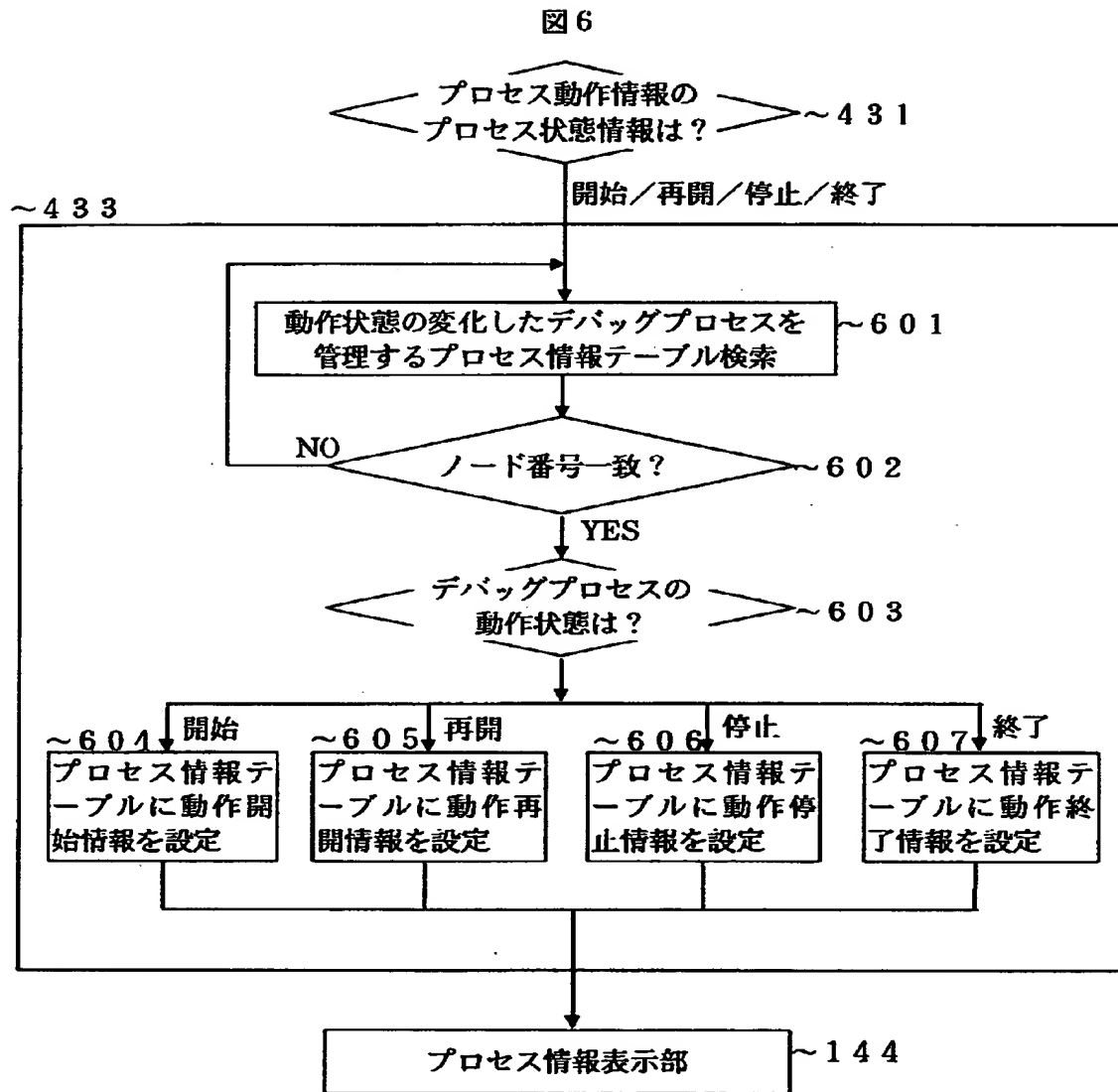
【図4】



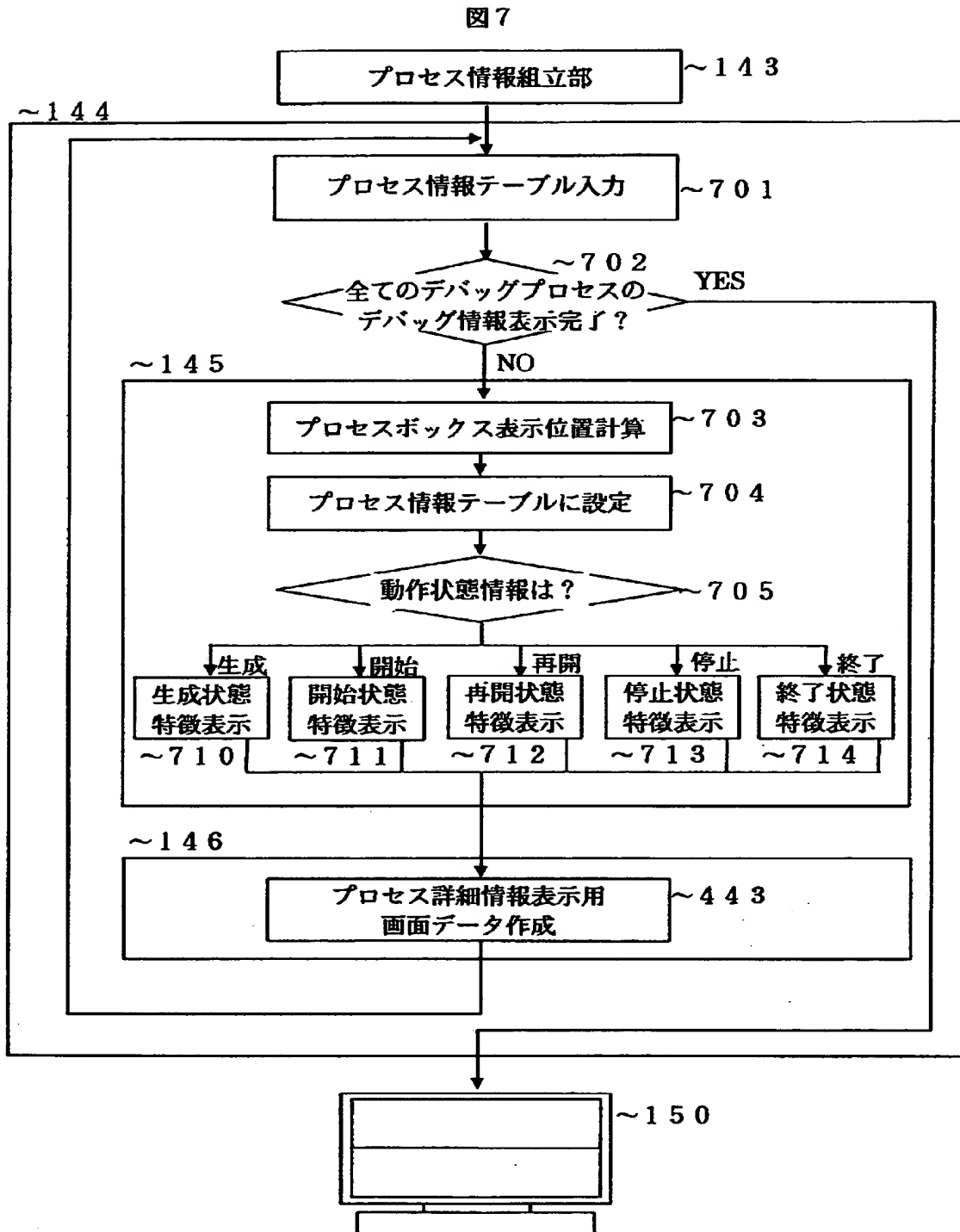
【図 5】



【図 6】

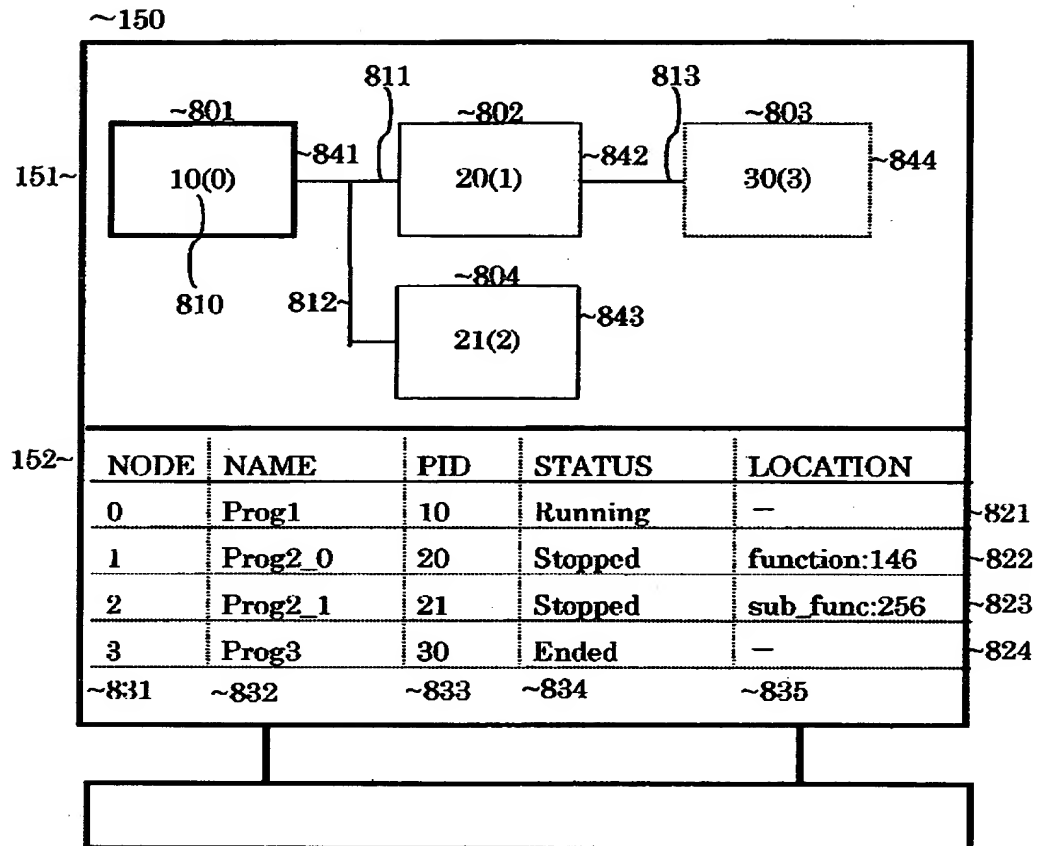


【図 7】

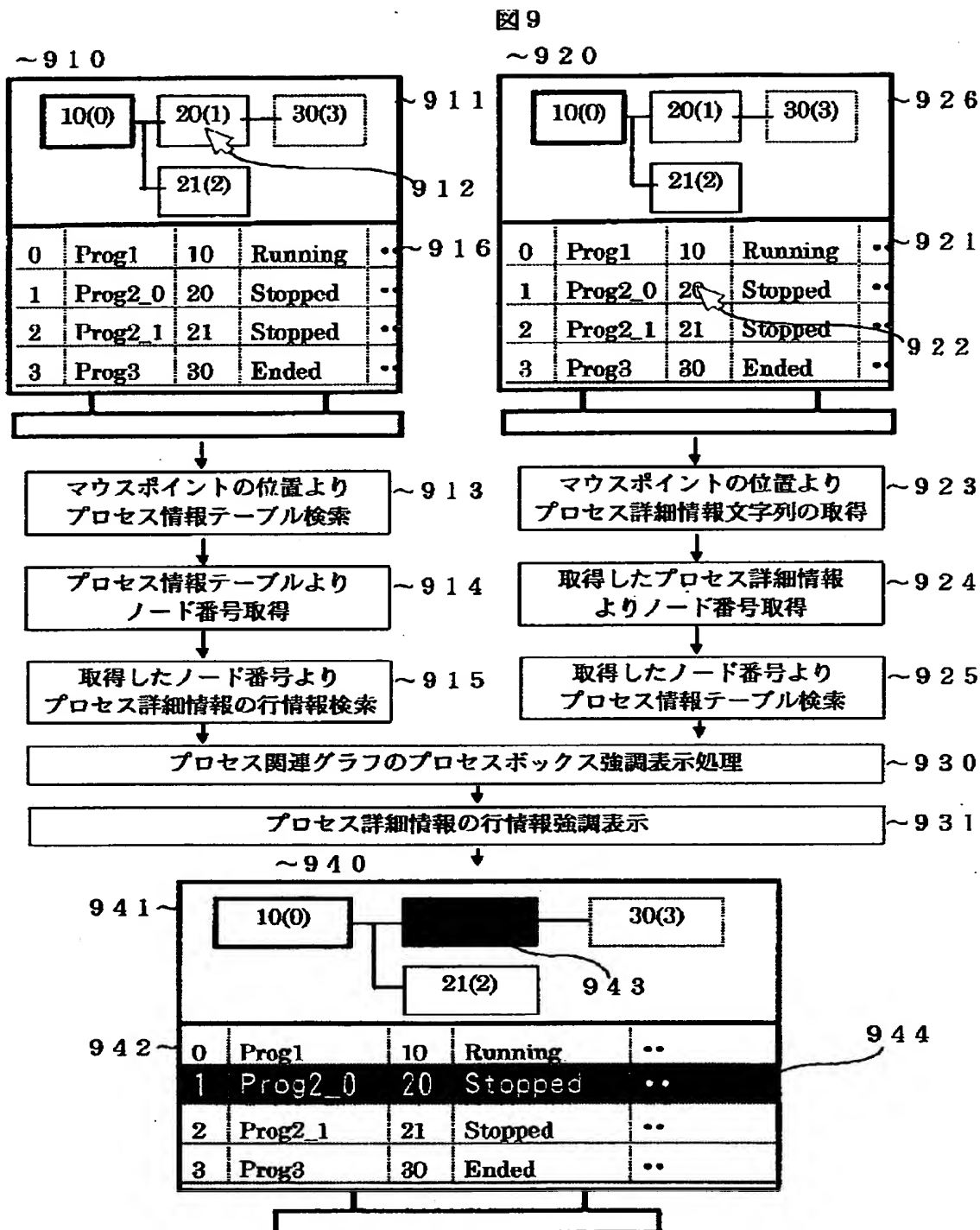


【図 8】

図 8



【図9】



BEST AVAILABLE COPY

【書類名】 要約書

【要約】

【課題】

マルチプロセスプログラムを構成する複数のプロセスの動作状態に関連付けたデバッグ情報を、プログラムの動作と連動しながら特徴表示するデバッグ支援装置を提供する。

【解決手段】

デバッグプログラム 1 0 0 を構成する複数のデバッグプロセス 1 0 1 の動作状態の変化を、デバッガ実行処理部 1 2 1 で監視し、デバッグプロセス 1 0 1 の動作状態に変化が生じた場合、プロセス動作情報出力部 1 2 2 はプロセス動作情報 1 3 1 を出力する。デバッガ情報表示部 1 4 0 は、プロセス動作情報を入力 1 4 2 し、デバッグプロセス 1 0 1 の親子、及び兄弟関係を視覚的に表現し、プロセスボックスを特徴表示する、プロセス関連グラフ 1 5 1 と、デバッグプロセス 1 0 1 の動作状態を詳細に表示するプロセス詳細情報 1 5 2 を同一画面に表示する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地

氏 名 株式会社日立製作所